Лабораторная работа №2. Условный оператор.

Теория 1. Неполный условный оператор

Формат записи:

```
if [условие]:
    [действие, если условие выполняется]
```

Примечание 1. Условие может принимать только два значения: истина (True) или ложь (False).

Примечание 2. Отступы в Python являются неотъемлемой частью кода. После служебного слова `if` блок кода информирует интерпретатор Python, как действовать, если условие истинно, и как – если оно ложно. Согласно синтаксису команды, после условия должно стоять двоеточие. Тело команды записывается в следующей строке. По соглашению PEP 8, для отступа блоков кода используются *4 пробела* (кнопка Tab). В Google colab используется 2 пробела.

Операторы сравнения.

Операторы сравнения	Выражение	Описание
`> (больше)`	`if x > 7`	если x больше 7
`< (меньше)`	`if x < 7`	если x меньше 7
`>= (больше или равно)`	`if x >= 7`	если x больше либо равен 7
`<= (меньше или равно)`	`if x <= 7`	если x меньше либо равен 7
`== (равно)`	`if x == 7`	если x равен 7
`!= (не равно)`	`if x != 7`	если x не равен 7

Все операторы, состоящие из двух знаков пишутся слитно, без пробелов между ними.

```
'''
Пример 2.1.1) Программа проверяет, какой язык
программирования
изучают пользователи и выдает положительный ответ, в случае,
когда введено: «Python»
'''
answer = input('Какой язык программирования мы изучаем?: ')
if answer == 'Python':
    print('Верно! Мы ботаем Python =)')
    print('Python - отличный язык!')
```

```
'''
Пример 2.1.2) Программа проверяет выполнение условия
и, если оно истинно, выводит на экран соответствующий
текст.
'''
x = int(input('Введите целое число x: '))
if x > 100000:
    print('x больше 100000')
    print('x не маленькое!')
if -100 < x <= 10:
    print('x больше -100 и меньше или равно 10')
if x == 0:
    print('x равно 0')
if x != 5:
    print('x не равно 5')
```

Задача 2.1. Расчет некоторого показателя производится при помощи заданной ниже функции. Причем исследователя интересуют лишь положительные значения. Напишите программу, которая позволяет их найти. Т.е., если значение функции в заданной точке (которая вводится с клавиатуры) больше нуля, то вывести его на экран в виде строки:
f([значение x])=[полученное значение f].
Результат вывести с числом знаков после запятой, равным Вашему варианту.
Пример вывода:
если введено x = -6.56, на экран должно быть выведено (для В-2):
f(-6.56) = 2277.40.
Подсказка: учтите, что x может быть любым действительным; вспомните, какими знаками обрамляются переменные для вывода на экран их значений, как убрать лишние пробелы, а также форматированный вывод действительных чисел.

В-1. $f(x)=4x^2-7$;
В-2. $f(x)=-8x^3+19$;
В-3. $f(x)=3x-11x^2$;
В-4. $f(x)=2x^4+1,7$;
В-5. $f(x)=-2x^{23}-1$;
В-6. $f(x)=11x^2-7$;
В-7. $f(x)=x-7x^2-12x^3$;
В-8. $f(x)=13x^{11}+12$;
В-9. $f(x)=-7x^2-7$;
В-10. $f(x)=20-x^5$;
В-11. $f(x)=12x^2+12$;

B-12. $f(x)=9x^9-1$;
B-13. $f(x)=19x^9+17$;
B-14. $f(x)=41+x^3-7x$;
B-15. $f(x)=16x^4+21$;
B-16. $f(x)=17x^{10}-71$;
B-17. $f(x)=-23-3x^7$;
B-18. $f(x)=-4+x^4-x^2$;
B-19. $f(x)=5x^2-10x+15x^{12}$;
B-20. $f(x)=100x^{14}+222$.

Теория 2. Классический условный оператор

Формат записи:

```
if [условие]:
   [действие, если условие выполняется]
else:
   [действие, если условие не выполняется]
```

Условный оператор такого вида позволяет выполнять одни действия, при истинности условия, другие – при его ложности.

```
'''
Пример 2.2.1) Программа проверяет, выполнено ли условие и
выводит на экран соответствующий текст.
'''
dose = int(input('Угадайте дозировку лекарства. Введите
числовое значение: '))
if dose != 500:
    print('Нет, мне назначена другая дозировка. Вы не
угадали!')
else:
    print('Вы угадали! Именно 500 мг')

'''
Пример 2.2.2). Программа сообщает здоров пользователь или
нет,
в зависимости от введенной температуры (упрощенный вариант,
не
предполагающий ввода аномальных значений).
'''
T = float(input('Введите свою температуру: '))
if T < 37:
    print('Вы абсолютно здоровы!')
else:
    print('Вы больны. Срочно обратитесь к врачу!')
```

Задача 2.2. Если значения выражений в точке совпадают, то вывести текст «Значения равны», и напечатать «Значения не равны», - в противном случае.

В-1. $f(x)=x+4$ и $f(x)=100x^{14}+222$;
В-2. $f(x)=-8x^3+19$ и $f(x)=7-x$;
В-3. $f(x)=34+x$ и $f(x)=-4+x^4-x^2$;
В-4. $f(x)=3x-6$ и $f(x)=-23-3x^7$;
В-5. $f(x)=17x^{10}-71$ и $f(x)=0{,}15x-11$;
В-6. $f(x)=11x^2-7$ и $f(x)=15-9x$;
В-7. $f(x)=41+x^3-7x$ и $f(x)=24-x^2$;
В-8. $f(x)=19x^9+17$ и $f(x)=13+12x$;

В-9. $f(x)=9x^9-1$ и $f(x)=3x-4$;
В-10. $f(x)=12x^2+12$ и $f(x)=4x+x^2$;
В-11. $f(x)=1,7x-3,1$ и $f(x)=20-x^5$;
В-12. $f(x)=-7x^2-7$ и $f(x)=5-7,1x$;
В-13. $f(x)=3+x$ и $f(x)=13x^{11}+12$;
В-14. $f(x)=16x^4+21$ и $f(x)=-2x+18$;
В-15. $f(x)=x-x^2$ и $f(x)=x-7x^2-12x^3$;
В-16. $f(x)=2,6x-92$ и $f(x)=-2x^{23}-1$;
В-17. $f(x)=2x^4+1,7$ и $f(x)=13x+3,8$;
В-18. $f(x)=3+2x$ и $f(x)=3x-11x^2$;
В-19. $f(x)=12-x$ и $f(x)=5x^2-10x+15x^{12}$;
В-20. $f(x)=4x^2-7$ и $f(x)=5,2x-3$.

Теория 3. Полный условный оператор

Формат записи:

```
if  [условие 1]:
    [действие, если условие 1 выполняется]
elif условие 2:
    [действие, если не выполняется условие 1, но выполняется
условие 2]
elif [условие 3]:
    [действие, если не выполнены условие 1 и условие 2, но
выполняется условие 3]
...
else:
    [действие, если ни одно условие не было выполнено]
```

Примечание1. Различных условий может быть любое разумное количество.

Примечание2. Полный условный оператор применяется в том случае, когда нельзя написать несколько независимых операторов if из-за пересечения условий, поскольку в таком случае будет напечатано сразу несколько значений вместо одного. Например, если условие1: x>5, а условие2: x>100, то при x>100 будут выполняться оба условия и производиться соответствующие действия).

```
'''
Пример 2.3.1) Программа вычисляет значение функции y в точке
x, если:
```

$$y=\begin{cases}7x, & x<-3;\\ 18x^{11}, & -3<x\le 5;\\ \dfrac{1-x}{x^2-4}, & x>5;\end{cases}$$

```
'''
x = float(input())
if x < -3:
    print('y =', 7 * x)
elif x == -3:
    print('функция не определена')
elif x <= 5:
    print('y =', 18 * x**11)
else:
    print('y =', (1 - x)/(x**2 - 4))
```

```
'''
Пример 2.3.2) Программа относит пользователя к определенной
возрастной категории по классификации ВОЗ.
'''
age = int(input('Введите свой возраст: ')
if age < 0 or age > 150:
    print('Возраст введен неверно')
elif age < 25:
    print('Детский возраст')
elif age < 44:
    print('Молодой возраст')
elif age< 60:
    print('Средний возраст')
elif age< 75:
    print('Пожилой возраст')
elif age< 90:
    print('Старческий возраст')
else:
    print('Долгожитель')
```

Задача 2.3. Найдите значение функции в точке x, введенной с клавиатуры. Результат выводите в виде соответствующего действительного числа или надписи «функция не определена».

В-1.
$$y=\begin{cases} x, & x<0; \\ x^3, & 0<x\le 2; \\ \dfrac{1}{2-x}, & x>2; \end{cases}$$

В-2.
$$y=\begin{cases} x-5, & x<-2; \\ x^3-\dfrac{1}{x}, & -2<x<0; \\ \dfrac{1}{7+x}, & x\ge 0; \end{cases}$$

B-3.
$$y=\begin{cases}6x+\dfrac{1}{x}, & x<-4;\\ 7-8x^2, & -4\le x<2;\\ \dfrac{5}{x-2}, & x>2;\end{cases}$$

B-4.
$$y=\begin{cases}2, & x<-2;\\ 4-x^2, & -2\le x<2;\\ x-2, & x>2;\end{cases}$$

B-5.
$$y=\begin{cases}\dfrac{1}{x}, & x<1;\\ x^4+5, & 1<x\le 2;\\ 8{,}1-x, & x>2;\end{cases}$$

B-6.
$$y=\begin{cases}\dfrac{2}{1-x^2}, & x<-1;\\ x+2x^2, & -1<x\le 0;\\ 1-\dfrac{x}{9}+\dfrac{2}{x}, & x>0;\end{cases}$$

B-7.
$$y=\begin{cases}2x, & x<-2;\\ x^3-5, & -2<x\le 3;\\ \dfrac{3-x}{4x}, & x>3;\end{cases}$$

B-8.
$$y=\begin{cases}\dfrac{11x}{3-x^2}, & x<-4;\\ 3+x^2, & -4<x\le 3;\\ \dfrac{x}{1+x}, & x>3;\end{cases}$$

B-9.
$$y=\begin{cases}\dfrac{1}{x^2}, & x<0;\\ 3x+16, & 0\le x<2;\\ 5-x^2, & x\ge 2;\end{cases}$$

B-10.
$$y=\begin{cases}7x, & x<-6;\\ \dfrac{x+7}{x}, & -6\le x<-2;\\ 1-x^3, & x\ge -2;\end{cases}$$

B-11.
$$y=\begin{cases}\dfrac{x+1}{x^2-1}, & x<-1;\\ 2x, & -1\le x<1;\\ -\dfrac{1}{2}, & x>1;\end{cases}$$

B-12.
$$y=\begin{cases}5x^3, & x<-8;\\ \dfrac{1}{x-4}, & -8<x\le -5;\\ 15+7x, & x>-5;\end{cases}$$

B-13.
$$y=\begin{cases}x^5, & x<-5;\\ \dfrac{2}{x-6}, & -5<x\le 3;\\ 3x, & x>3;\end{cases}$$

B-14.
$$y=\begin{cases}\dfrac{2x^5}{4-x}, & x<4;\\ 5x, & 4<x\le 7;\\ x^2, & x>7;\end{cases}$$

B-15.
$$y=\begin{cases}\dfrac{3x^4-2}{5-x}, & x<-7;\\ 5x-2, & -7<x\le 7;\\ 4x^2+22, & x>7;\end{cases}$$

B-16.
$$y=\begin{cases}8, & x<-3;\\ \dfrac{5x}{x-2}, & -3<x\le 1;\\ x^8, & x>1;\end{cases}$$

B-17.
$$y=\begin{cases}\dfrac{x}{5}, & x<0;\\ \dfrac{11}{x+2}, & 0<x\le 5;\\ x^7+7, & x>5;\end{cases}$$

B-18.
$$y=\begin{cases}x^3, & x<-1;\\ \dfrac{x}{x+1}, & -1<x\le 1;\\ 2x^6+5x, & x>1;\end{cases}$$

B-19. $$y=\begin{cases}\dfrac{2}{x^3}, & x<-3;\\ 2, & -3<x\le 2;\\ x+\dfrac{6}{x}, & x>2;\end{cases}$$

B-20. $$y=\begin{cases}x^3, & x<1;\\ 2, & 1<x\le 2;\\ 3x, & x>2.\end{cases}$$

Теория 4. Логические операторы

Когда требуется учесть несколько условий, написание условного оператора можно упростить с помощью логических операторов:

and – логическое умножение;

or – логическое сложение;

not – логическое отрицание.

Оператор `and`.

Как работает: все условия должны выполняться одновременно.

```
'''
Пример 2.4.1) Программа, как доктор-недоучка, по наличию у
пациента ровно двух симптомов одновременно определяет
заболевание.
'''
symptom1, symptom2 = input('Есть ли у Вас головная боль?
Введите: да или нет\n'),input('Есть ли у Вас температура?
Введите: да или нет\n ')
if symptom1 == 'да' and symptom2 == 'да':
# если оба симптома присутствуют одновременно, то
- Covid-19
    print('У Вас Covid-19')
else:
    print('Вы абсолютно здоровы')
```

Примечание. Как и в операторе print(), если мы работаем с текстовыми переменными, необходимо заключать текст в кавычки, двойные или одинарные.

Таблица истинности для оператора `and`:

X	Y	X and Y
False	False	False
False	True	False
True	False	False
True	True	True

Выражение истинно, когда оба условия X и Y истинны.

Оператор `or`.

Как работает: должно выполняться хотя бы одно из условий.

```
'''
Пример 2.4.2) Программа, как доктор-недоучка, по
наличию у пациента хотя бы одного симптома определяет
заболевание.
'''
symptom1, symptom2 = input('Есть ли у Вас головная боль?
Введите: да или нет\n '),input('Есть ли у Вас температура?
Введите: да или нет\n ')
if symptom1 == 'да' or symptom2 == 'да':
# если присутствует хотя бы один, то - Covid-19
     print('У Вас Covid-19')
else:
    print('Вы абсолютно здоровы')
```

Таблица истинности для оператора or .

X	Y	X or Y
False	False	False
False	True	True
True	False	True
True	True	True

Выражение истинно, когда хотя бы одно из условий X или Y истинно.

Оператор `not`.

Как работает: инвертирует (т.е. заменяет на противоположный) результат логического выражения.

Таблица истинности для оператора not :

X	not X
False	True
True	False

Выражение истинно, когда условие X истинно.

Примечание. Логические операторы можно комбинировать.

```
'''
Пример 2.4.3) Программа, как доктор-недоучка, по
наличию небольшого набора симптомов определяет заболевание.
'''
symptom1, symptom2, symptom3 = input('Есть ли у Вас головная
боль? Введите: да или нет\n '),input('Есть ли у Вас
температура? Введите: да или нет\n '), input('Есть ли у Вас
кашель? Введите: да или нет\n ')
if not symptom1 == 'нет' and not symptom2 == 'нет'  or
symptom3 == 'да':
# если есть кашель или одновременно головная боль и
температура, то - Covid-19
    print('У Вас Covid-19')
else:
    print('Вы абсолютно здоровы')
```

Примечание. Условие `not symptom1 == 'нет'`, разумеется, равносильно: `symptom1 == 'да'`. В примере просто демонстрируется применение отрицания.

Приоритеты логических операторов

not ⟶ and ⟶ or

Для **явного указания порядка** выполнения условных операторов **используют скобки**.

Задача 2.4. Проверьте: различны или нет все цифры в числе, составляющем размер вашей ладони в миллиметрах.

Задача 2.5. Напишите программу, которая по тому или иному сочетанию симптомов s1, s2, s3 и s4 определяет наличие болезни. Заболевания и их симптомы можно найти по справочнику болезней, например, на https://spbpmc.ru/reference/directory/. Выберите любое заболевание, имеющее требуемое количество основных симптомов (обратите при этом внимание на наличие отрицания в выражении – симптомы, которых нет подберите для другого заболевания).

Примечание. Высказывания в вариантах составлены случайным образом. Не пытайтесь подобрать полностью подходящее к ним заболевание. Выбирайте любое, лишь бы для него можно было указать нужное количество симптомов.

В-1. не s1 и s2 и (s3 или не s4);

В-2. s1 или (не s2 или s3) или s4;

В-3. s1 и (s2 или не s3 или s4);

В-4. (не s1 или не s2) и s3 или s4;
В-5. не s1 и s2 и (s3 или s4);
В-6. s1 и (s2 и s3 или не s4);
В-7. (s1 или s2 и s3) и не s4;
В-8. (не s1 и s2 или s3) или s4;
В-9. не s1 и (s2 или не s3) и s4 ;
В-10. не s1 и (s2 или s3 и s4);
В-11. (s1 или s2) и s3 или не s4;
В-12. (не s1 и s2 или s3) и не s4 ;
В-13. s1 или s2 и (не s3 или s4);
В-14. s1 и не s2 и (s3 или s4);
В-15. не s1 или не s2 и (s3 или s4);
В-16. s1 и (s2 и не s3 или не s4);
В-17. (не s1 или s2) и (s3 или не s4);
В-18. (s1 и s2 или s3) и не s4;
В-19. не s1 и (s2 или s3) или s4;
В-20. не s1 и (s2 или s3 или s4).

Задача 2.6. Напишите программу, вычисляющую значение выражения u = (x + y)·z, если введенные с клавиатуры x, y, z удовлетворяют заданным ниже условиям, и выводящую «Введено некорректное значение», если иначе.

В-1. $x \in [2,8; 3)$ и $y \in (4; 9,1]$ или $z \notin [-1; 7)$;
В-2. $(x \in (-2; 0)$ или $y \notin [3; 6,7])$ и $z \in (-6; 1]$;
В-3. $x \in (-5; 1]$ или $y \in (-4,9; 4)$ и $z \notin [-3; 2]$;
В-4. $x \notin [0; 6)$ и $(y \in (2; 3,6]$ или $z \notin (-3; 0])$;
В-5. $x \in [8; 9,2)$ и $y \notin [2; 6]$ или $z \notin (-3; 8]$;
В-6. $(x \notin [-5; 0,5)$ или $y \in (3; 8])$ и $z \in (-7; 7)$;
В-7. $x \in (-2; 3)$ или $y \notin [-8,8; 4]$ и $z \in [-1; 1]$;
В-8. $x \in (0,1; 6]$ и $(y \notin [2; 5]$ или $z \notin (-3; 4])$;
В-9. $x \notin [1,7; 2)$ и $y \in (-2; 6]$ или $z \in [5; 7]$;
В-10. $(x \notin [-5; 0,5)$ или $y \in (-2; 5])$ и $z \in [-7; 7]$;
В-11. $x \in [5,2; 14)$ или $y \notin (-11; 67]$ и $z \notin [-14; -1]$;
В-12. $x \notin [0; 6,3]$ и $(y \in (-5; 4]$ или $z \notin [3; 12])$;
В-13. $x \notin (7; 22,6]$ и $y \in [-5; -4]$ или $z \notin (3; 23)$;
В-14. $(x \notin (-14; 5]$ или $y \in (-67,8; 9])$ и $z \notin [-2; 91]$;
В-15. $x \notin [3; 11)$ или $y \notin [-1,7; 6)$ и $z \in [-1; 0]$;
В-16. $x \in (0; 4)$ и $(y \in [-5,9; 2]$ или $z \notin [-1; 2])$;
В-17. $x \in [-13; 10]$ и $y \in (-5; 4,7]$ или $z \notin [7; 156]$;
В-18. $(x \notin [-1; 15]$ или $y \in [-8; 13))$ и $z \in [-2,9; 9]$;
В-19. $x \in (3; 18]$ или $y \notin [-1; 0]$ и $z \in (-5; 0,3]$;
В-20. $x \notin [-8; -3)$ и $(y \in (-7,7; 4]$ или $z \notin [-4; 10])$;

Теория 5. Функции min() и max(). Функция abs()

Функции min() и max().

Для определения минимального или максимального значения в Python применяются функции `min()` и `max()` соответственно.

Примечание1. Аргументы функций `min()` и `max()` должны иметь один тип.

Примечание2. Количество аргументов не ограничено.

```
# Пример 2.5.1) Нахождение минимального и максимального
значений из набора данных исследования.
a = max(3, 8, -3, 12, 9, 0, 11, 5, 5, -1)
b = min(3.14, 2.17, 5.8, 1.82, 3.54, 2.26, 2.91)
print(a)
 print(b)
```

Функция abs()

Модуль (абсолютная величина) числа a обозначается $|a|$:

$$|a| = \begin{cases} a, & a \geq 0; \\ -a, & a < 0. \end{cases}$$

Для нахождения модуля числа в Python существует функция `abs()`.

```
#Пример 2.5.2) Вычисление модуля числа.
 print(abs(10))
print(abs(-7))
print(abs(0))
print(abs(-17.67))
```

Задача 2.7. Напишите программу, которая находит наименьшее из 4-х цифр, составляющих год Вашего рождения, если в этом году Вам исполняется/исполнилось 19 лет и наибольшее из них, если нет. С клавиатуры вводится лишь одно число – год вашего рождения и автоматически проверяется, удовлетворяет ли возраст указанному условию, и на экран выводится либо минимум, либо максимум, в зависимости от полученного результата.

Задача 2.8. Напишите программу, которая для 3-х, введенных с клавиатуры действительных чисел a1, a2, a3, вычисляет:

В-1. значение |a1| + |a2| : |a3|, если все они меньше числа (-10,1), но не меньше числа (-15), иначе выводит |a1| · |a2| · |a3|.

В-2. значение |a1| : |a2| · |a3|, если все они меньше числа (-5,5), но не меньше числа (-13), иначе выводит |a1| – |a2| · |a3|.

В-3. значение (–|a1| – |a2|) : |a3|, если все они больше 11,6, но не больше 17, иначе выводит |a1| – |a2| + |a3|.

В-4. значение (|a1| – |a2|) · |a3|, если все они больше (-13), но не больше 20,8, иначе выводит |a1| – |a2| : |a3|.
В-5. значение |a1| · |a2| + |a3|, если все они меньше числа 1,7, но не меньше (-1), иначе выводит |a1| : |a2| · |a3|.
В-6. значение |a1| : |a2| + |a3|, если все они больше числа 14,8, но не больше 21, иначе выводит |a1| · (–|a2|) + |a3|.
В-7. значение |a1| · (–|a2|) + |a3|, если все они больше числа (-1,2), но не больше 25, иначе выводит (|a1| – |a2|) : |a3|.
В-8. значение |a1| : (–|a2| + |a3|), если все они меньше числа 15,4, но не меньше (-5), иначе выводит |a1| – (–|a2|) · |a3|.
В-9. значение (–|a1|) : (|a2| + |a3|), если все они меньше 0, но не меньше числа (-20,2), иначе выводит |a1| – (–|a2| · (–|a3|)).
В-10. значение |a1| : (–|a2|) – |a3|, если все они больше 1, но не больше 5,2, иначе выводит (|a1| – |a2|) · (–|a3|).
В-11. значение |a1| + (–|a2|) · |a3|, если все они больше (-7,1), но не больше 7, иначе выводит (|a1| + |a2|) : (–|a3|).
В-12. значение |a1| – |a2| : |a3|, если все они меньше (-3), но не меньше числа (-11,6), иначе выводит |a1| + |a2| · (–|a3|).
В-13. значение (–(|a1| – |a2|)) : |a3|, если все они меньше числа (-0,1), но не меньше числа (-20), иначе выводит |a1| – |a2| · |a3|.
В-14. значение (–|a1| – |a2| – |a3|), если все они больше числа (-8,8), но не больше 100, иначе выводит (–(|a1| + |a2| : |a3|)).
В-15. значение (–(–|a1| : |a2| – |a3|), если все они больше 6, но не больше числа 90,9, иначе выводит (–(|a1| + |a2| : |a3|)).
В-16. значение |a3| : (–|a1| – |a2|), если все они меньше числа 2, но не меньше числа 0,7, иначе выводит (–|a1| + |a2| · |a3|).
В-17. значение |a3| · (|a1| – |a2|), если все они меньше числа 9,9, но не меньше числа (-23), иначе выводит (–|a1| : |a2| · |a3|).
В-18. значение (|a1| + |a3|) · |a2|, если все они больше числа (-2), но не больше числа 36,6, иначе выводит |a3| : (|a1| + |a2|).
В-19. значение (|a1| – |a3|) : |a2|, если все они больше числа 7, но не больше числа 68,9, иначе выводит |a3| – (|a1| · |a2|).
В-20. значение |a3| · (–|a1| – |a2|), если все они меньше числа 5, но не меньше числа (-8,3), иначе выводит (–|a2| + |a1| : |a3|).

Задача 2.9. Напишите программу, которая запрашивает у больного результат диагностики и определяет, принадлежит ли введенное число промежутку [2; 6).

Задача 2.10.Напишите программу, которая запрашивает у больного результат диагностики и определяет, принадлежит ли введенное число объединению промежутков (-∞; 4]∪(7;+ ∞).

Задача 2.11. Напишите программу, которая запрашивает у больного результат диагностики и определяет, принадлежит ли введенное число объединению промежутков (0; 1]∪[1,7; 20].

Задача 2.12. Для проведения диспансеризации отбираются школьники в возрасте от 14 до 17 лет (включительно). По введенным дню, месяцу и году рождения определить необходимость прохождения диспансеризации (количество полных лет определяется на текущую дату).

Задача 2.13. Группу добровольцев, собранную для испытания нового лекарственного средства, разбили на три, с номерами, последней цифрой в которых, является номер группы: 1, 2 или 3. Напишите программу, которая по введенному идентификационному номеру определяет, к какой из подгрупп относится испытуемый.

Задача 2.14. Напишите программу, которая дает оценку суточного ритма артериального давления (АД) по степени снижения его ночного суточного индекса (СИ): СИ = 100% х (АДд – АДн)/АДд, где АДд – среднее АД в период бодрствования, АДн – среднее АД в период сна. Т.е., в зависимости от величины СИ для введенных с клавиатуры АДд и АДн, выводит на экран характер ночного снижения АД и тип суточной кривой АД с пояснениями, в соответствии с таблицей:

Характер ночного снижения АД	Тип кривой АД	Суточный индекс,%
нормальное	диппер	10-20
недостаточное	нон-диппер	0-10
ночная гипертония	найт-пикер	менее 0
чрезмерное	овер-диппер	более 20

Задача 2.15. Напишите программу, которая вычисляет значение функции y по введенному с клавиатуры значению x:

$$y = \begin{cases} |3x-1|, & x < -6; \\ 4 + минимум(-5x; 3x), & -6 \le x \le 10; \\ 2x^2 + 9|x| - 7, & x > 10. \end{cases}$$

Задача 2.16. Три болезни имеют схожие симптомы. Если у больного присутствуют симптомы s1, s2, s3, возраст больше 40 лет и вес больше 80 кг, то выставляется диагноз A; если при тех же симптомах возраст больше 50 лет и вес больше 90 кг и пациент курит, то это уже диагноз B и при возрасте больше 80 лет,

весе больше 95 кг и дополнительном симптоме s4 – диагноз С. Написать программу, которая определяет диагноз пациента по введенным параметрам или печатает «Здоров», если иначе.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.

1. «Поколение Python»: курс для начинающих // https://stepik.org/course/58852/promo (дата обращения 22.11.22).
2. Академия искусственного интеллекта для школьников https://ai- academy.ru/ (дата обращения 22.03.22).
3. Беккер, М.С. Применение математических методов в медицине: методическое пособие / М.С. Беккер. – Ессентуки, 2006. – 35 с.
4. 12. Все о Python. Программирование на Python 3 // https://all- python.ru/ (дата обращения 22.03.22)
5. Биофармация: учебник для фармацевтических вузов и факультетов / В.В. Гладышева [и др.]; под ред. В.В. Гладышева. – 2-е изд. –. Днипро: ЧМП «Экономика». – 2018. – 250 с.
6. Павленко, В. Интерактивный учебник языка Python / В. Павленко, В. Соломатин, Д.П. Кириенко, команда Pythontutor // https://pythontutor.ru/lessons/sets/(дата обращения 22.11.22).
7. Культин, Н.Б. С/С++ в задачах и примерах / Н.Б. Культин. – 2-е изд., перераб. и доп. – Спб.: БХВ-Петербург, 2011. – 368 с.
8. Медицинский справочник онлайн // http://www.makhaon.com/index.php?lng=ru&p=dict (дата обращения 22.11.22).
9. Питонтьютор // https://pythontutor.ru/ (дата обращения 22.11.22).
10. Питончик // https://pythonchik.ru/ (дата обращения 22.11.22).
11. Показатели СМАД в практической и исследовательской кардиологии // https://www.schiller.ru/profile/articles/smad/146/ (дата обращения 22.11.22).
12. Беккер, М.С. Применение математических расчетов в медицине: методическая разработка / М.С. Беккер // https://nsportal.ru/shkola/algebra/library/2012/06/09/primenenie-matematicheskikh-raschetov-v-meditsine (дата обращения 22.11.22).
13. Набиуллина С.Н. Решение расчетных задач на разведение антибиотиков: методическая разработка фрагмента занятия по математике / С.Н. Набиуллина. – Курган, 2018 // https://www.informio.ru/files/main/documents/2018/12/fragment_zanjatija_Nabiullina_SN.docx (дата обращения 22.11.22).
14. Лукашевский, С. Руководство по языку Python / С. Лукашевский // https://pyprog.pro/python/py/py_guide.html (дата обращения 22.11.22).
15. Справочник заболеваний // https://illness.docdoc.ru/ (дата обращения 22.11.22).